



TP 2 : Java Serveur Page

1) Lecture de données de formulaire à l'aide de JSP

Sur vos machines, un serveur d'applications Tomcat est installé. Ce serveur possède un conteneur JEE permettant la réalisation de projets JEE.

- Tout d'abord, vérifiez que Tomcat est correctement installé sur vos machines et configuré avec Eclipse

Lecture de données de formulaire à l'aide de JSP

JSP gère automatiquement l'analyse des données de formulaire à l'aide des méthodes suivantes, selon la situation :

- **getParameter ()** - Vous appelez la méthode **request.getParameter ()** pour obtenir la valeur d'un paramètre de formulaire.
- **getParameterValues ()** - Appelez cette méthode si le paramètre apparaît plusieurs fois et renvoie plusieurs valeurs, par exemple une case à cocher.
- **getParameterNames ()** - Appelez cette méthode si vous souhaitez une liste complète de tous les paramètres de la demande en cours.
- **getInputStream ()** - Appelez cette méthode pour lire le flux de données binaires provenant du client.

2) Exemple de méthode GET à l'aide d'une URL

L'URL suivante transmettra deux valeurs au programme HelloForm à l'aide de la méthode GET.

http://localhost:8080/main.jsp?first_name=YASMINE&last_name=AMINE

Vous trouverez ci-dessous le programme JSP **main.jsp** permettant de gérer les données saisies par le navigateur Web. Nous allons utiliser la méthode **getParameter()** qui facilite l'accès aux informations transmises – Complétez le code suivant afin de récupérer les valeurs des IDs du lien, exécuté et interprété le résultat:

```
<html>
  <head>
    <title>Using GET Method to Read Form Data</title>
  </head>

  <body>
    <h1>Using GET Method to Read Form Data</h1>
    <ul>
      <li><p><b>First Name:</b>
        <%= request.getParameter("first_name")%>
      </p></li>
      <li><p><b>Last Name:</b>
        <%= request.getParameter("last_name")%>
      </p></li>
    </ul>

  </body>
</html>
```

Maintenant, tapez http://localhost:8080/main.jsp?first_name=YASMINE&last_name=AMINE dans la zone de votre navigateur.



3) [Exemple de méthode GET à l'aide d'un formulaire](#)

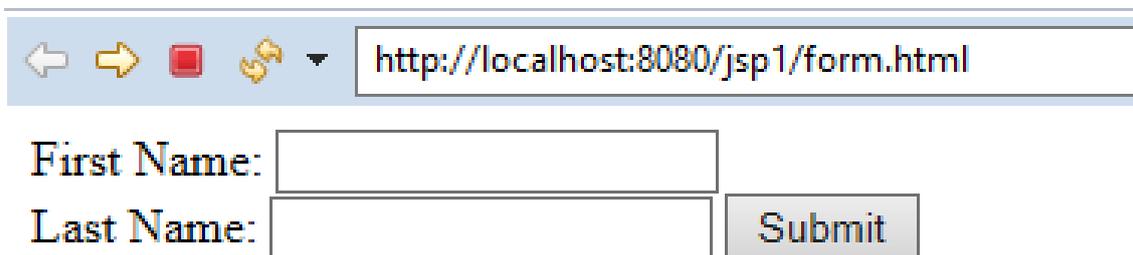
L'exemple ci-dessous transmet deux valeurs à l'aide du formulaire `form.html` et du bouton d'envoi. Nous allons utiliser le même fichier JSP `main.jsp` pour gérer cette entrée.

```
<html>
  <body>

    <form action = "main.jsp" method = "GET">
      First Name: <input type = "text" name = "first_name">
      <br />
      Last Name: <input type = "text" name = "last_name" />
      <input type = "submit" value = "Submit" />
    </form>

  </body>
</html>
```

form.xml



Essayez de saisir le prénom et le nom de famille, puis cliquez sur le bouton Soumettre pour afficher le résultat sur votre ordinateur local où Tomcat est en cours d'exécution. Sur la base de l'entrée fournie, il générera un résultat similaire à celui mentionné dans l'exemple précédent.

4) Transmission de données de case à cocher au programme JSP

Les cases à cocher sont utilisées lorsque plusieurs options doivent être sélectionnées.

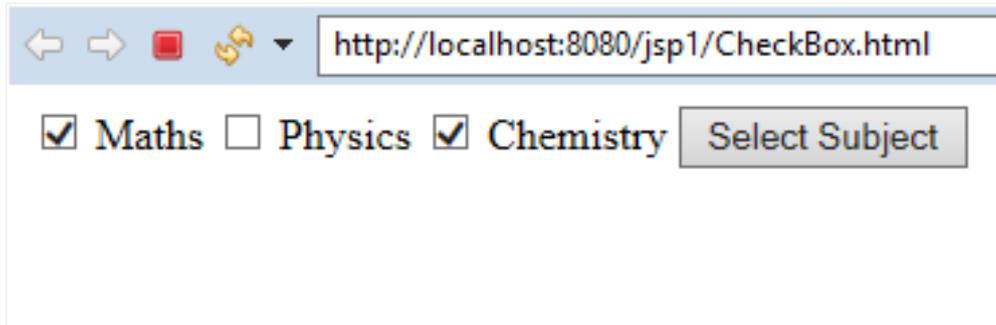
Voici un exemple de code HTML, `CheckBox.htm`, pour un formulaire avec deux cases à cocher.

```
<html>
  <body>

    <form action = "main.jsp" method = "POST" target = "_blank">
      <input type = "checkbox" name = "maths" checked = "checked" /> Maths
      <input type = "checkbox" name = "physics" /> Physics
      <input type = "checkbox" name = "chemistry" checked = "checked" /> Chemistry
      <input type = "submit" value = "Select Subject" />
    </form>

  </body>
</html>
```

Le code ci-dessus générera le résultat suivant :



← → 🛑 🔄 ▾ http://localhost:8080/jsp1/CheckBox.html

Maths Physics Chemistry

Voici le programme `mainCheckBox.jsp` JSP permettant de gérer l'entrée donnée par le navigateur Web pour le bouton de case à cocher.

```
<html>
  <head>
    <title>Reading Checkbox Data</title>
  </head>

  <body>
    <h1>Reading Checkbox Data</h1>

    <ul>
      <li><p><b>Maths Flag:</b>
        <%= request.getParameter("maths")%>
      </p></li>
      <li><p><b>Physics Flag:</b>
        <%= request.getParameter("physics")%>
      </p></li>
      <li><p><b>Chemistry Flag:</b>
        <%= request.getParameter("chemistry")%>
      </p></li>
    </ul>

  </body>
</html>
```

Le programme ci-dessus générera le résultat suivant :



5) Lecture de tous les paramètres de formulaire

Voici un exemple générique qui utilise la méthode `getParameterNames()` de `HttpServletRequest` pour lire tous les paramètres de formulaire disponibles. Cette méthode retourne une énumération qui contient les noms de paramètre dans un ordre non spécifié.

Une fois que nous avons une énumération, nous pouvons boucler celle-ci de la manière habituelle, en utilisant la méthode `hasMoreElements()` pour déterminer quand arrêter et en utilisant la méthode `nextElement()` pour obtenir chaque nom de paramètre.

Voici le programme `mainCheckBoxAll.jsp` permettant de gérer l'entrée donnée par le navigateur Web pour le bouton de case à cocher.



```
<%@ page import = "java.io.*,java.util.*" %>

<html>
  <head>
    <title>HTTP Header Request Example</title>
  </head>

  <body>
    <center>
      <h2>HTTP Header Request Example</h2>
      <table width = "100%" border = "1" align = "center">
        <tr bgcolor = "#949494">
          <th>Param Name</th>
          <th>Param Value(s)</th>
        </tr>
        <%
          Enumeration paramNames = request.getParameterNames();
          while(paramNames.hasMoreElements()) {
            String paramName = (String)paramNames.nextElement();
            out.print("<tr><td>" + paramName + "</td>\n");
            String paramValue = request.getHeader(paramName);
            out.println("<td> " + paramValue + "</td></tr>\n");
          }
        %>
      </table>
    </center>

  </body>
</html>
```